(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: **0 254 812 A2**

(12) EUROPEAN PATENT APPLICATION

(21) Application number: 87105550.5

(51) Int. Cl.4: **H04L 9/00**

(22) Date of filing: 14.04.87

(30) Priority: 27.06.86 US 879784

(43) Date of publication of application:
03.02.88 Bulletin 88/05

(84) Designated Contracting States:
DE FR GB IT

(71) Applicant: **International Business Machines Corporation**
Old Orchard Road
Armonk, N.Y. 10504(US)

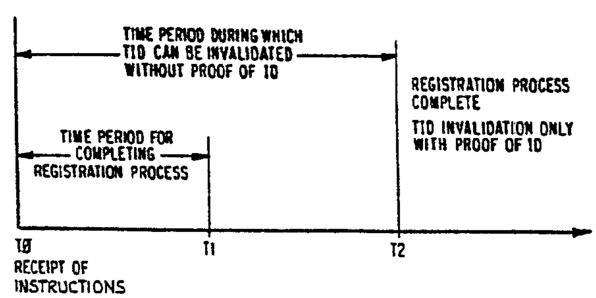
(72) Inventor: **Matyas, Stephen Michael**
8978 Miles Place
Manassas, Virginia 22110(US)

(74) Representative: **Grant, Iain Murray**
IBM United Kingdom Limited Intellectual Property Department Hursley Park
Winchester Hampshire SO21 2JN(GB)

(54) Cryptovariable initialisation in a public key network.

(57) A procedure is disclosed for initialising with security and integrity a large number of terminals in an EFT/POS network with cryptographic variables. Each terminal in the network is provided with a terminal identification known to the key distribution centre, the public key of the key distribution centre is stored in the cryptographic facility of each terminal. A terminal initialiser is designated for each terminal, and the terminal initialiser for each terminal is notified of two expiration times for the purposes of registering the terminal's cryptovariable with the key distribution centre. The cryptovariable is generated by the terminal using its cryptographic facility. Prior to the first expiration time, a registration request is prepared and transmitted to the key distribution centre. The registration request includes the terminal identification and the cryptovariable. When the key distribution centre receives this request, the cryptovariable is temporarily registered and that fact is acknowledged to the requesting terminal. After the expiration of the second time, the registration is complete. Provisions are also made for invalidating a terminal identification in the event that more than one registration is attempted for a given terminal identification or that the registration was not made in time. The same procedure can be used to initialise cryptovariables of users of a network. The protocol is basically the same except that a user identification is used instead of a terminal identification, and the users may be provided with a transportable media, such as a magnetic stripe card or the like, which stores the user cryptovariable and can be read by terminals in the network.

FIG.2



EP 0 254 812 A2

CRYPTOVARIABLE INITIALISATION IN A PUBLIC KEY NETWORK

The present invention relates to cryptovariable initialisation in a public key network.

Cryptographic methods are required whenever information must be sent securely through an uncontrolled, possibly hostile, environment such as a communications network. To employ these methods requires the initialisation of system nodes with cryptographic variables, i.e. cryptographic keys. Initialisation procedures are required for networks with large numbers, perhaps hundreds of thousands, of terminals in locations with low physical security. Such networks are typified by modern Electronic Funds Transfer (EFT) and Point of Sale (POS) networks such as those used by interstate banking and retailing establishments.

Normally, security personnel are employed to initialise a system with cryptographic keys. In an implementation using a symmetric algorithm, such as the Data Encryption Algorithm (DEA), trusted personnel must handle secret keys. In an implementation using an asymmetric algorithm such as the Rivest, Shamir, Adleman algorithm (RSA), trusted personnel might also initialise the system with secret keys, although here it is possible to require only public keys to be distributed. An asymmetric algorithm can also be referred to as a Public Key Algorithm (PKA). In the latter approach, the required secret key and corresponding public key are generated internally by the node. The secret key is stored whereas the public key is displayed so that it can be distributed by security personnel to the appropriate node communicating with the terminal, and, for this, a channel with integrity is required since otherwise a fake public key, corresponding to a fake secret key, might be accepted by the authenticating node.

As the number of terminals in a network grows, one might expect that a certain economy-of-scale would come into play to reduce the overall cost of distributing keys using security personnel (e.g. couriers). However, current projections indicate quite the opposite, and it is expected that the cost of key distribution using couriers will grow at least proportionally with the number of terminals, and perhaps even more. Although it is true that travel distances between sites serviced by such couriers might well be reduced as more and more nodes are located within a given geographical area, the increased work in scheduling and coordinating courier visits at these sites would undoubtedly more than offset this expected advantage. Another major difficulty would involve smaller networks jointing with larger networks, or joining of several small networks into one large network. The problem is that a small network joining a large network might well find that key distribution is not limited to only the small network. It might require key distribution across the total network.

An early approach to key distribution, the courier-based distribution of secret keys is well known in the art. One of the first proposals for handling key distribution with a PK algorithm simply involved the exchange of public keys over a communication channel by a pair of devices wishing to communicate. This technique, however, lacked integrity since it was possible for an adversary to pose as a genuine node merely by sending his public key to another party in the network. An adversary could also perform an active attack against two devices intending to enter into a communication by intercepting the exchanged public keys and forwarding his own public key to the respective devices. This would allow the adversary to intercept, decrypt, read, and re-encrypt all communications from one device to another.

Another proposal for distributing public keys was to register them with a key distribution centre. Anyone wishing to communicate with a particular party would first contact with KDC to obtain a copy of that party's public key. To provide integrity, the KDC would prepare a short message containing the public key and the ID of the device or user to which the key belonged and the "sign" this message by decrypting the message using the secret key of the KDC. In advance, the public key of the KDC would be distributed to each node or device in the system, which could then be used to validate the message containing the public key and signature received from the KDC by encrypting the received signature with the public key. This then provided a path with integrity to distribute the public keys of each user or device. There remains, however, the issue of integrity of the keys during registration. As the initial registration process could merely consist of communicating the public key to the KDC in some sort of message saying that this is my key and please register it, an adversary could falsely register a public key in the name of someone else.

Racal-Milgo has implemented a method of key distribution via a PK algorithm. Aware of the potential for spoofing, Racal-Milgo implemented an anti-spoofing procedure involving a telephone call in which the parties verified their respective public keys by comparing verification information calculated on the public keys. Briefly, two parties who wish to communicate each generate a public key/secret key pair and then exchange their public keys via the communication channel. Upon receipt

of the public key, each party calculates a prescribed function of the public key. The parties then contact each other via telephone and exchange the calculated values, which are then verified by the originating parties. If the correct values are communicated, then each party has received the correct public key. This procedure is described in an article entitled "Public Key Security" by C. R. Abbruscato published in the December 1984 issue of Telecommunications. The weak link in this anti-spoofing defence is that the telephone channel itself must have integrity or the callers must recognise each other's voice.

Bell Telephone Laboratories has described a similar technique for anti-spoofing which pre-dates the Racal-Milgo technique. Bell's technique involves verifying the public keys by calculating and mailing the key validation information to the originating node instead of communicating the information by voice over a telephone communication channel. Otherwise, the concept is the same. The procedure is described in an article by Frank H. Myers entitled "A Data Link Encryption System", NTC Conference Record, National Telecommunications Conference, Washington, D.C., Nov. 27-29, 1979. Again this anti-spoofing defence requires that the postal system handling mail have integrity, else the anti-spoofing check could again be spoofed.

In a recent paper, Carl H. Meyer and Stephen M. Matyas describe a method of key installation/distribution. See "Installation and Distribution of Cryptographic Variables in an EFT/POS Network with a Large Number of Terminals", Proceedings of SECURICOM 86 (1986). This approach calls for the installation of the secret terminal keys at a trusted node, for example, a Key Distribution Centre (KDC). The terminals with their installed keys are then transported to their final destinations. The secret initial terminal key is protected during transport (initial key distribution) by a secure hardware design defined a Tamper Resistant Module (TRM) which appears to achieve the highest degree of key protection. This procedure has the same degree of complexity and security whether the employed cryptographic algorithm is symmetric (e.g. the Data Encryption Algorithm (DEA)) or asymmetric (e.g. the Rivest, Shamir, Adleman algorithm (RSA)).

Another approach allows the initial secret terminal keys to be distributed with key mailers, similar to the procedures used to distribute Personal Identification Numbers (PINs). Although this approach of initial key distribution is less costly and does not require terminal initialisation at a trusted node, it is less secure since it is easier to obtain secret information by intercepting mail than by attacking a TRM design.

With the the trend toward networks with hundreds of thousands of terminal devices, the need for cost effective, practical, and secure techniques for the distribution of cryptographic keys poses a special challenge to the designer of a cryptographic system. The need for secret cryptographic keys at each system node can be demonstrated by one of the more important security requirements; i.e. the requirement to assure unaltered transmission of messages between network nodes. If this requirement is satisfied, it is said that the messages have integrity. To achieve this requires the introduction of cryptographic error detection codes. Such a code must be a function of the message and a secret quantity such that even a minute change in the message will have a corresponding change in the code. A secret quantity is required in generating this code in order that only the owner of such a secret key can generate a valid quantity. (The cryptographic concepts used applies to a check for message integrity as well as to assure the integrity and authenticity of other entities such as system nodes, cryptographic keys and system users.) Consequently, the requirement to provide message integrity checks dictates the installation of secret keys in all system nodes.

Another problem posed by large networks is how to provide sufficient message security, integrity as well as secrecy, in an environment where the network entry point devices have low physical security. A low cost POS terminal installed in a supermarket is a good example of this. In such an environment, it would be unwise to store a secret key in terminals that would compromise network security beyond that of the single terminal, should the key become compromised. Also it would be advantageous if the compromise of such a key would not allow an adversary to decrypt previously transmitted and intercepted data.

The anticipated cost and other associated problems with courier-based key distribution in networks with very large numbers of nodes has caused a heightened concern to find better, less expensive, and equally secure methods of key distribution than traditional courier-based methods.

As further background to the present invention, the reader may make reference to U.S. Patents No. 4,200,770 to Hellman et al, No. 4,218,582 to Hellman et al and No. 4,405,829 to Rivest et al for discussions of public key algorithms. Also of interest is U.S. Patent No. 4,206,315 to Matyas et al which describes on column 4, line 62, to column 6, line 17, the generation of a message and signature. U.S. Patent No. 4,386,234 to Ehrsam et al describes at column 5, lines 26 to 42, a terminal with an integrated security device. This device is the cryptographic facility which is described by Meyer and Matyas in Cryptography: A New Dimension in

Computer Data Security, John Wiley & Sons (1982), at pages 222 to 226. U.S. Patent No. 4,238,853 to Ehrsam et al shows in Figure 9 and describes in the text on column 20, lines 49 to 68, and column 21, lines 1 to 9, a procedure for generation of random numbers which can be used by the host data security device (i.e. cryptographic facility) as part of a process of generating cryptographic keys. In the same patent, at column 4, lines 54 to 68, and column 5, lines 1 to 51, there is a description of the host data security device and the key generation process.

It is therefore an object of the present invention to initialise with security and integrity a large number of nodes in an information handling system with cryptographic variables without requiring couriers to transport these cryptographic variables.

The present invention provides a method of initialising cryptovariables in a multiterminal network which uses a public key algorithm and wherein the number of terminals is not essentially limited, the method comprising

establishing a key distribution centre within the network;

generating a public and secret key pair for the key distribution centre;

providing a cryptographic facility in each terminal that is intended to be secure;

storing the public key of the key distribution centre in each such cryptographic facility;

providing and storing at the key distribution centre, for each authorised entity - user or terminal - an individual identification; and

informing an initialiser - the user or an appointee in the case of a terminal - of two expiration times for the purposes of registering a cryptovariable with the key distribution centre in order that the initialiser can generate a cryptovariable at a or the terminal, using the cryptographic facility and transmit over the network to the key distribution centre a registration request message including the identification and the cryptovariable pair before the first of the expiration times;

the key distribution centre temporarily registering the cryptovariable against the identification, if received before the first expiration time and, if any further communication is received before the second expiration period in respect thereof, cancelling the registration, whereas, after the second expiration period, only cancelling the registration upon proof of authorisation.

Initialisation of a terminal is performed by a designated representative called the terminal initialiser, while initialisation of the network to accept a user will be expected,to be performed by the user. In all cases, the terminal initialiser is a person who will act responsibly to carry out the steps of the terminal initialisation procedure, such procedure comprising the steps of the terminal initialisation procedure, such procedure comprising the steps of causing the terminal to generate and register one or more cryptovariables with a designated Key Distribution Centre (KDC) and promptly reporting to the KDC any encountered problems. Typically, the terminal initialiser will be an employee of the organisation at the location where the terminal is physically installed, such as a terminal user, terminal owner, manager, or member of the local site security. In situations where a third party key distribution centre is employed, the terminal initialiser may be a locally appointed agent of the KDC. The terminal initialiser has no responsibility for transporting keys, public or private, or for installing secret keys by entering them directly into a cryptographic device. Therefore, the terminal initialiser is not a courier, and does not perform the functions of a courier.

According to one embodiment of the invention using a public key algorithm, each terminal in the network is provided with a cryptographic facility (CF) consisting of hardware and software components that perform the necessary cryptographic functions to support the required cryptographic operations. A subset of these functions support the terminal initialisation procedure of the invention. Overall cryptographic security, including that of the terminal initialisation procedure, rests on an assumption of integrity of the CF, including stored keys and programs, and associated supporting software, which is guaranteed by the design and by other physical security measures instituted by the user. Prior to the terminal initialisation procedure according to the invention, the KDC generates a public key and secret key pair (PKkdc, SKkdc), which are the keys that operate with the public key algorithm. A unique nonsecret Terminal Identifier (TID) and the public key of the Key Distribution Centre (PKkdc) are assumed to have been installed in the CF of the terminal. The PKkcd could be installed, for example, in microcode as part of the manufacturing process of the terminal. Alternatively, it could be installed at a central location and the

terminals with the installed PKkdc and TID shipped to the final destination, or it could be installed by the terminal initialiser, that is, locally after the terminal has been installed.

For each terminal which is to be initialised, as previously mentioned, the KDC designates a terminal initialiser who is responsible for carrying out the necessary terminal initialisation procedure at that device. Each terminal initialiser is provided with a set of instructions outlining the terminal initialisation procedure. The security of the procedure rests on the assumption that the terminal initialiser will comply with the issued instructions and understands that failure to comply with these instructions may result in an adversary successfully registering a key with the key distribution centre. The KDC also provides to the terminal initialiser two expiration dates, ordinarily separated by several days, which delimit periods of time in which certain prescribed steps within the terminal initialisation procedure must be completed. The security of the procedure rests on the assumption that the terminal initialiser receives notifications of the two expiration dates and the terminal initialisation instructions at some time well in advance of the expiration dates so that the steps of the procedure can be performed within the prescribed allotted time.

According to the terminal initialisation procedure of the invention, prior to the first expiration date, a cryptovariable can be temporarily registered at the KDC under the designated TID provided that the TID has not been invalidated and no other prior cryptovariable has been registered for that TID. In the specific embodiment disclosed herein, the cryptovariable registered is a public key, and therefore this process is called "public key registration". If a public key has already been registered under a given TID, an attempt to register a different public key under that same TID will result in the TID being invalidated. After the first expiration date, the "public key registration" process is disabled at the KDC for that TID.

Prior to the second expiration date, the KDC permits a TID to be invalidated without "proof" of the identity of the requestor. This process is called "ID invalidation without proof of identity". After the second expiration date, the process of "ID invalidation without proof of identity" is disabled for that TID, and the temporary status of the registration is considered changed to that of a permanent registration.

After the second expiration date, the KDC permits a TID to be invalidated only after the requestor has been identified and authenticated and his or her authorisation to invalidate that TID has been verified. This process is called "ID invalidation with proof of identity".

After the second expiration date and upon request, the KDC will issue a PK Certificate for any TID provided that the TID is valid and a public key has been registered for that TID. A PK Certificate consists of a TID, public key, certificate expiration date, other data, and a digital signature produced on the foregoing data using the secret key of the KDC. One recommended method for calculating a signature is to first calculate an intermediate value or function of the message, using a strong one-way cryptographic function. This intermediate value is then decrypted with the secret key SKkdc to produce the signature. U.S. Patent No. 4,206,315 describes, for example, a signature technique which calculates the signature on a one-way function of the message instead of the message itself. U.S. Patent No. 4,405,829 describes the process of encryption and decryption with the so-called RSA public key algorithm. If the TID is invalid or no public key has been registered, an appropriate response message is prepared on which a digital signature is calculated using the secret key of the KDC and the message and signature are returned to the requesting terminal.

Under normal operating conditions, the terminal initialisation procedure proceeds as follows. Well in advance of the first expiration date, a public key and secret key pair are generated at the terminal using an available key generation procedure. A public key registration request message containing the TID and public key of the terminal is sent to the KDC. Under normal conditions no adversary will have interfered with the process, and therefore no public key will yet be registered under the designated TID.

Therefore, the KDC registers the public key under the specified TID, prepares an appropriate response message containing the TID and public key on which a digital signature is calculated using the secret key of the KDC in the manner previously described, and the message and signature are returned to the requesting terminal. After authenticating the received message, the requesting device signals the terminal initialiser that the desired public key has been temporarily registered at the KDC under the specified TID. The procedure for authenticating a signature is similar to the procedure for calculating a signature. The same intermediate one way function of the message, which was used in calculating the signature, is again calculated from the message. The signature is then encrypted using the public key of the KDC (PKkdc), and the recovered one way function of the message is compared for equality with the calculated one way function of the message. If the

comparison is favourable, the message and signature are accepted; otherwise, if the comparison is unfavourable, the message and signature are rejected.

The protocol now requires a delay, and the terminal initialiser must now wait for the passage of the second expiration time in order that the KDC may assure that the temporarily registered public key is genuine; i.e. that it originated from the authorised, appointed terminal initialiser. After the second expiration time, a terminal-initialiser-initiated message containing the TID is sent to the KDC requesting "ID Verification" for that TID. Under normal conditions no adversary will have interfered with the process and therefore the specified TID will be valid and the previously temporarily registered public key will still be registered, but due to the expiration of the second time period, registration is now considered permanent. Therefore, the KDC prepares and returns a message to the requesting terminal specifying the registered PK for that TID. A digital signature is prepared on this message using SKkdc which allows the requesting terminal to authenticate the received message using the installed PKkdc in the manner previously described. This signals satisfactory completion of the terminal initialisation procedure and provides the necessary proof that the desired public key has been successfully registered at the KDC. Alternatively, the KDC could return a PK certificate to the requesting terminal, and this would also serve as proof to the terminal that the public key had been registered.

Once an authenticated response has been received from the KDC stating that a public key has been temporarily registered or that the TID has been invalidated, the worst that could happen is than an adversary could cause a genuine temporarily registered public key to be erased by invalidating the TID prior to the second expiration time. Hence, for practical purposes, a "safe" state is reached, and it is therefore possible with no loss in security to allow a protocol variation wherein the terminal-initialiser-initiated message sent to the KDC requesting "ID verification" following the second expiration time can be replaced by a similar terminal-user-initiated message. This protocol variation has the advantage that ordinarily the terminal initialiser can complete the terminal initialisation procedure with only one terminal initialisation status and the value of T2, completes the protocol after the second expiration time. Of course, the protocol variation is the same as the original protocol when the terminal initialiser and the terminal user are the same person.

In a network where it is convenient for the KDC to send messages to the terminals, such as in a store-and-forward electronic mail distribution system, yet another variation on the protocol is possible. The step following the second expiration time wherein a terminal-initialiser-initiated or terminal-user-initiated message is sent to the KDC requesting "ID verification" is replaced by a step wherein the KDC automatically prepares and sends a response to the original requesting terminal. This response is just the same as that which would have been sent in response to a request for "ID verification" except here the response is triggered by reaching the second expiration time rather than upon receiving a request message. Otherwise, the protocol is the same. If no response is received at the terminal within a reasonable period of time after the second expiration time, the terminal initialiser or the terminal user, depending on which protocol is used, reports this discrepancy to the KDC.

The present invention will be described further by way of example with reference to embodiments thereof as illustrated in the accompanying drawings, in which:

FIGURE 1 is a block diagram of an EFT/POS network showing the initial states of the key distribution centre and several terminals, with a terminal initialiser for each terminal;

FIGURE 2 is a time line diagram showing in graphical form the rules for the public key registration of a terminal in the system shown in Figure 1;

FIGURE 3 is a state diagram showing the possible states of a terminal in the terminal initialisation procedure; and

FIGURE 4 is a state diagram showing the possible states of the KDC in the terminal initialisation procedure.

Referring now to the drawings, and more particularly to Figure 1, there is shown in simplified block diagram form an EFT/POS network which includes a Key Distribution Centre (KDC) 10 and a two EFT/POS terminals 12 and 14 of what may in fact be hundreds of thousands of terminals in the network. There is installed a Public Key, PKkdc, and a Secret Key, SKkdc, at the KDC 10. PKkdc is also installed in each of the terminals 12 and 14, and each terminal is provided with its own Terminal Identification (TID) number. The generation of PKkdc, SKkdc at the KDC and the installation of PKkdc and TID at the terminal is well within the current state of the art. It is recommended, although not required, that the assigned TIDs constitute a sparse set (e.g. by some random or arbitrary selection from among a very large set of possible numbers). This would diminish interfer-

ence and disruption from common "hacker"-related attacks. The notion is that even though the TIDs are nonsecret, there is no reason to make them easy to guess.

Terminal initialisers 16 and 18 are assigned to each terminal 12 and 14, respectively. Each terminal initialiser 16 and 18 is provided with a set of terminal initialisation instructions 20 and 22, respectively. The terminal initialisers do not necessarily receive their instructions on the same date. These instructions may include two expiration times, T1 and T2, or the two expiration times could be separately communicated to the terminal initialisers. Ordinarily, T1 and T2 would be different among the terminals, but they might be the same for an installation grouping. For example, terminal initialiser 16 might receive his instructions on January 1, 1986. and be assigned expiration times of 12 o'clock midnight on January 15, 1986, for T1 and 12 o'clock midnight on January 22, 1986, for T2. Terminal initialiser 18, on the other hand, might receive his instructions on February 5, 1986, and be assigned expiration times of 12 o'clock midnight on February 19, 1986, for T1 and 12 o'clock midnight on February 26, 1986, for T2. Denoting the date of receipt of the installation instructions as T0, the rules for the public key registration with respect to the expiration times T1 and T2 are shown in Figure 2.

The terminal initialisation procedure comprises a series of steps that must be performed by the designated terminal initialiser. The terminal initialiser is a person who can be counted on to act responsibly to carry out the steps of the terminal initialisation procedure. Typically, the terminal initialiser will be an employee of the organisation at the location where the terminal is physically installed, such as a terminal user, terminal owner, manager, or member of the local site security. As mentioned, the KDC provides the designated terminal initialiser with two expiration dates, T1 and T2. As illustrated in Figure 2, the first of these dates, T1, represents a cutoff date by which the public key portion of the terminal initialisation procedure must be completed, and the second of these dates, T2, defines the expiration of a grace period during which the terminal initialiser must report to the KDC if he or she has not received from the KDC an authenticated, positive notification that the requested public key for the requested TID has been temporarily registered or the requested TID has been invalidated. As mentioned, the terminal initialiser receives T1 and T2 with the initialisation instructions or T1 and T2 could be separately communicated to the terminal initialiser. The public mail system or an internal company-wide private mail system could be used to distribute the terminal initialisation instructions.

The terminal initialisation instructions will now be described with reference to the state diagrams shown in Figures 3 and 4 of the drawings. Reference may also be had to Figure 2 which shows the time line diagram that graphically illustrates the results of the initialisation procedure according to the invention.

In step 1 of the initialisation procedure, the terminal initialiser is reminded that he or she should have received two notification letters from the key distribution centre. Each notification specifies that as the assigned terminal initialiser for the designated terminal, a set of terminal initialisation instructions should be received by a specified date. The instructions specify what the terminal initialiser is to do in the event that the corroborating notices were not received. The instructions also contain certain vital information, some of which may be optional information, such as the name of the terminal initialiser, device ID, first and second expiration dates and times, which the terminal initialiser must check for reasonableness. If the current date and time is beyond the first expiration date and time, but not beyond the second expiration date and time, the procedure begins with to Step 5 of the instructions to invalidate the TID. This is represented in the state diagram of Figure 3 as a change from state 1 to state 2. If the current date and time is beyond the second expiration date and time, the procedure begins with Step 6 where proof of identity will be required to invalidate the TID. This is represented in Figure 3 as a change in the state diagram from state 2 to state 3. These special conditions will be discussed in more detail later with respect to the referenced steps. For the time being, it will be assumed that T1 has not yet expired.

In step 2, the terminal initialiser causes the generation of a terminal public key and secret key pair using the terminal's cryptographic facility. This step is part of a procedure referred to as "personalisation" and is preparatory to the public key registration.

In step 3, the terminal initialiser performs the public key registration with the key distribution centre, provided that the current date and time is prior to the first expiration time, T1, as indicated in Figure 2. This is accomplished by sending a PK registration request message containing the terminal ID (TIC), device public key, and time-variant data to the KDC. A telephone number may be dialled as one way to establish a connection with the key distribution centre. The terminal then sends the prepared PK registration request message to the KDC. In response to a received PK registration request message, the KDC prepares an appropriate response message reflecting the state of the KDC and the outcome of processing the PK registration

request message, the public key registered or temporarily registered for the requested TID, the requested TID, plus the same time-variant data received in the request message. A digital signature is calculated on the response message using the secret key, DKkdc, of the KDC and using the procedure for calculating signatures previously described, and the message and signature are sent to the requesting terminal. If a response message and signature are received from the KDC, the terminal authenticates the message and signature using the KDC public key, PKkdc, in the manner previously described.

Several outcomes are possible for the actions initiated in the public key registration process:

(a) a connection could not be made with the KDC,

(b) no response was received from the KDC,

(c) the response received from the KDC was invalid or insufficient to determine the state of the KDC,

(d) a valid response was received from the KDC indicating that the KDC is in state A and no action has been taken by the KDC,

(e) a valid response was received from the KDC indicating that the KDC has transitioned from state A to state B and that the requested public key has been temporarily registered,

(f) a valid response was received from the KDC indicating that the KDC is in state C and that a public key different from the requested public key is already registered, or

(g) a valid response was received from the KDC indicating that the KDC is in state D, i.e. the TID in the public key registration request message has been or was previously invalidated at the KDC.

If outcome a, b or c occurs, the registration status will be uncertain. In that case, and also for outcome d which indicates that no public key has yet been temporarily registered, the terminal initialiser may try again. If outcome e occurs, the procedure requires that the terminal initialiser wait for the expiration of the second expiration time and then verify that the terminal's public key has been registered (Step 4). This outcome is indicated by the state change from state 1 to state 4 in the state diagram of Figure 3 and the state change from state A to state B in the state diagram of Figure 4. If outcome f occurs, it signifies to the terminal initialiser the second expiration time passed and that a potentially different public key has already been registered under the TID specified in the public key registration request message. In this case, the terminal initialiser must invalidate the TID using a procedure which requires proof of identity (Step 6). This outcome is indicated by the state change from state 1 to state 3 in the state diagram of Figure 3. If outcome g occurs, the TID is invalidated, and no further action is required. This outcome is indicated by the state change from state 1 to state 7 in the state diagram of Figure 3 and either the state change from state B to state D or that the original state D remains unchanged in the state diagram of Figure 4.

In Step 4, the terminal initialiser performs a verification that the terminal's public key has been registered at the key distribution centre following the expiration of the second expiration time or, more generally, to query the status of the KDC at any time. To perform this step, it is not necessary that the terminal initialiser be located at the designated terminal as any terminal with a cryptographic facility that has been properly initialised will suffice. When performing the ID verification, an ID verification request message containing the TID and time-variant data is first prepared. Then a telephone number may be dialled to establish a connection with the KDC. In response to a received ID verification request message, the KDC prepares an appropriate response message reflecting the state of the KDC and the outcome of processing the ID verification request message. The response message contains the TID, optionally expiration times T1 and T2 for TID, terminal ID status at the KDC, public key registered or temporarily registered at the KDC, and the same time-variant data received in the ID verification request message. A digital signature is calculated on the response message using the secret key, SKkdc, of the KDC in the manner previously described, and the message and signature are sent to the requesting terminal. If a response message and signature are received from the KDC, the device authenticates the message and signature using the KDC public key, PKkdc, again in the manner previously described.

Several outcomes are possible for the actions initiated in the ID verification process:

(a) a connection could not be made with the KDC,

(b) no response was received from the KDC,

(c) the response from the KDC was invalid or insufficient to determine the state of the KDC,

(d) a valid response was received from the KDC indicating that the KDC is in state A, i.e. the first expiration time has not yet expired and a public key has not yet been temporarily registered,

(e) a valid response was received from the KDC indicating that the KDC is in state B and that a public key whose value is specified has been temporarily registered,

(f) a valid response was received from the KDC indicating that the KDC is in state C and that a public key whose value is specified has been registered, or

(g) a valid response was received from the KDC indicating that the KDC is in state D, i.e. the TID has been invalidated.

The ID verification can be used at any point in the terminal initialisation process to determine the state of the KDC, although its primary purpose is to determine the state of the KDC after a public key has been temporarily registered and the second expiration time has expired. In that event, only two outcomes f and g are possible. Outcome f indicates that the temporarily registered public key now becomes permanently registered. No action is required; the terminal initialisation procedure has been completed. Outcomes a through g indicate no state changes in Figures 3 or 4.

Variations in the basic protocol are possible. Once an authenticated response has been received from the KDC stating that a public key has been temporarily registered or that the TID has been invalidated, the worst that can happen is that an adversary could cause a genuine temporarily registered public key to be erased by invalidating the TID prior to the second expiration time. Thus, it is possible with no loss of security to replace the terminal-initialiser-initiated message requesting "ID verification" to be replaced by a similar terminal-user-initiated message. Alternatively, the terminal-initialiser-initiated or the terminal-user-initiated message could be replaced by a step wherein the KDC automatically prepares and sends a response to the requesting terminal upon reaching the second expiration time.

The terminal initialiser would use Step 5 to invalidate the TID without proof of identity. This step is performed only when the current date and time is prior to the second expiration date, T2, as indicated in Figure 2. Again, the terminal initialiser does not need to be located at the designated terminal as any terminal in the network with an installed cryptographic facility which has been properly initialised will suffice. An ID invalidation request message containing the TID and time-variant data is first prepared. A telephone number may be dialled to establish a connection with the KDC. The terminal then sends the previously prepared ID invalidation request message to the KDC. In response to a received ID invalidation request message, the KDC invalidates the device ID provided that the current date and time is prior to the second expiration time, T2. The KDC then prepares an appropriate response message reflecting the state of the KDC and the outcome of processing the ID invalidation request message, plus it includes the same time-variant data received in the request message. A digital signature is calculated on the response message using SKkdc in the manner previously described, and the message and signature are sent to the requesting terminal. If a response message and signature are received from the KDC, the terminal authenticates the message and signature using PKkdc, in the manner previously described.

Again, several outcomes are possible:

(a) a connection could not be made with the KDC,

(b) no response was received from the KDC,

(c) the response received from the KDC was invalid or insufficient to determine the state of the KDC,

(d) a valid response was received from the KDC indicating that the KDC is in a state other than state D, or

(e) a valid response was received from the KDC indicating that the KDC is in state D.

Outcome a, b, c, or d indicates no state change in Figures 3 or 4. Outcome (e) is indicated in Figure 3 by the state changing from state 2 to state 7 and in Figure 4 by the state changing from state A to state D, from state B to state D, or that the original state D remains unchanged.

It is also possible to use an ordinary voice-grade telephone channel to request ID invalidation. The terminal initialiser selects and dials one of the numbers provided for the purpose and, when a connection is made, verbally requests that the TID be invalidated. The terminal initialiser will then be provided with an ID invalidation confirmation number if the ID invalidation request is accepted. The terminal initialiser must now authenticate the ID invalidation confirmation number, and this is done by using the ID invalidation confirmation number and the TID as inputs to a nonsecret one-way cryptographic function to calculate a reference number. This reference number is compared for equality with an ID invalidation number which is provided in the initialisation instructions or which may be separately communicated to the terminal initialiser; e.g. with the first and second expiration times T1 and T2. If the comparison is favourable, the terminal ID invalidation was successful. A simple and straight forward procedure for implementing the concept of an ID invalidation number is as follows. For each TID, the KDC randomly selects a unique 64-bit ID invalidation confirmation number. The 64-bit invalidation number is calculated by encrypting the TID using the ID invalidation confirmation number as a cryptographic key. Using the ID invalidation confirmation as a cryptographic key thus makes it impossible to reverse; i.e. given the TID and the ID invalidation number, it is impossible to derive the ID invalidation confirmation number. The so produced ID invalidation confirmation numbers are stored at the KDC indexed by TID, whereas the ID invalidation numbers are distributed to the terminal initialisers.

Step 6 is used for ID invalidation but with proof of identity to minimise potential disruption and denial of service. This step must be performed whenever the current date and time is past the second expiration time, T2, as shown in Figure 2. The verification portion of Step 6 requires that the terminal initialiser be located at a properly initialised terminal with an installed cryptographic facility.

Ordinarily, multiple options would be available for proving identity and invalidating a TID. However, time is of the essence. A fraudulently registered key should be invalidated within the system as quickly as possible. A separate authorisation channel to the KDC is the key element, and proof of identity could be based on a password, personal key, magnetic stripe card, voice recognition, or other techniques well established within the state of the art for proving identity. Another possibility involves a fact to face meeting with security personnel who are authorised to invalidate a registered key. Proof of identity could be based on personal recognition or identifying documents including the original initialisation instructions and letters of notification. An ID invalidation confirmation number is provided to the terminal initialiser using any convenient communication channel after the request for ID invalidation has been accepted by the KDC. The terminal initialiser authenticates the ID invalidation confirmation number using the same procedure described earlier under Step 5 for ID invalidation without proof of identity. The successful completion of this step is reflected in the state diagram of Figure 3 by the state changing from state 3 to state 7.

With more specific reference to Figure 3 of the drawings, there is first shown the state diagram of a terminal in a network for the procedure outlined above. Beginning at "Start", the public key registration process is invoked at state 1. Assuming first that the process is carried out normally, the terminal received from the KDC a message that the registration was accepted and this message is authenticated by the terminal. The state now becomes state 4 indicating that the requested public key has been temporarily registered at the KDC.

State 4 includes a delay or wait process which expires with T2, thus signalling the expiration of the period during which the TID can be invalidated without proof of the requestor's identity. At the expiration of T2, the state becomes state 5 which is the ID verification state. Two conditions are shown which cause no transition from state 5. One of these is a time out condition indicating that no response to an ID verification request message has been received form the KDC, and the other is an indication that a response to an ID verification request message has been received from the KDC but is insufficient to conclude that ID verification has been completed successfully. In either case, the terminal initialiser may retry the ID verification process. On the other hand, if a verification response message is received from the KDC indicating that the requested public key is permanently registered and the response is authenticated by the terminal, the state changes to state 6 providing an indication that the public key registration was completed successfully. Alternatively, the KDC may provide a response message to the terminal that the TID has been invalidated, in which case if the response message is authenticated the state changes to state 7 providing an indication that the public key registration was unsuccessful and that terminal initialisation will be carried out using a secondary or backup procedure.

Returning now to state 1, it will be observed that there are two conditions which cause no state transition out of state 1. The first of these is a time out condition indicating that no response was received from the KDC in response to the public key registration process. The second provides an indication that the response received is insufficient to conclude that either the public key registration was accepted or that the TID has been invalidated. In either case, the terminal initialiser may retry the public key registration process. Several conditions can cause a state change. If the public key registration response message is received from the KDC indicating that the TID has been invalidated and the response is authenticated by the terminal, the state changes to state 7 providing an indication that public key registration was unsuccessful and that terminal initialisation will be carried out via a secondary or backup procedure. If T1 expires before the public key registration is made with the KDC, then the state becomes state 2 where ID invalidation can be made without proof of identity. It is also possible to move to state 2 prior to the expiration of T1 if the terminal initialiser decides to invalidate the TID rather than continue with the public key registration process.

State 2 has two conditions which cause no state transition out of state 2. The first is a time out condition indicating no response has been received from the KDC in response to the ID invalidation procedure. The second indicates that a response was received from the KDC but is insufficient to conclude that the ID has been invalidated. In either case, the terminal initialiser may retry the ID invalidation procedure. On the other hand, if an ID invalidation response message is received from the KDC indicating that the TID has been invalidated and the response is authenticated by the terminal, the state changes to state 7. The transition to state 7 may also occur as a result of an oral communication in which an ID invalidation confirmation number

is received from the KDC and authenticated at the terminal. If the ID invalidation procedure has not been completed by the expiration of T2, then the state becomes state 3.

In state 3, ID invalidation can only be made with proof of identity. There are again two conditions which cause no state transition out of state 3. The first is a time out condition indicating no response has been received from the KDC in response to the ID invalidation procedure. The second indicates that a response was received from the KDC but is insufficient to conclude that the ID has been invalidated. In either case, the terminal initialiser may retry the ID invalidation procedure. On the other hand, if a response is received from the KDC indicating that the TID has been invalidated and the response is authenticated by the terminal, the state changes to state 7.

Figure 4 shows the state diagram for the KDC. Beginning with "Start" in state A, a registration of a public key for a requesting terminal identifier TID has not yet been received and accepted. As long as T1 has not expired, the KDC will accept a public key registration request but thereafter a registration request will not be accepted. If a valid public key registration request is received before T1, a transition from state A to state B occurs, which invokes the acceptance process for the registration process. This amounts to a temporary registration of the public key, wherein the received public key generated at the terminal is stored with the TID in a table at the KDC. The KDC then prepares a message including a signature on the message which is transmitted back to the requesting terminal to acknowledge the temporary registration of the public key. At the expiration of T2 with no intervening invalidation of the TID, a transition from state B to state C occurs, indicating that the temporarily registered public key is now permanently registered and that the registration process is complete. In this state, a public key certificate can be issued.

Returning to state A, if a public key registration request is received from a terminal but the TID or some other parameter in the request is invalid, the KDC does nothing; i.e. there is no acknowledging message sent to the terminal. If T1 expires without the public key registration process being completed, the state goes to state D where the TID is invalidated. The same result occurs from state B if before the expiration of T1 a registration request is received for the TID but for a different public key. In other words, there are two competing requests for public key registration using the same TID but for different public keys. In such a case, the state changes from state B to state D.

A transition from state A, B or C to state D also occurs as the result of a request for ID invalidation. A request for invalidation of ID without proof of identity, while in state A or B, will cause a transition to state D. If the public key registration is complete and the state is state C, an invalidation of the ID can be made only with proof of identity.

In the procedures discussed, communication with the KDC from a terminal in the network was made via a commercial telephone line. Preferably tight control measures would be employed over the dissemination and distribution of telephone numbers used with the procedure for reporting problems, even though the telephone numbers could never be considered secret. This would lessen interference and disruption from outsiders.

The preferred embodiment of the invention as described is then a procedure for initialisation of cryptographic variables in an EFT/POS network with a large number of terminals, perhaps numbering in the hundreds of thousands. The procedure recognises that the EFT/POS terminal must be small and inexpensive and will be installed in an insecure environment. In the procedure, the terminal is equipped with a cryptographic facility which it uses to generate a terminal public key and secret key pair. The generated public key is sent without integrity by transmitting it over an exposed channel, such as a commercial telephone line. The KDC accepts the public key of the terminal and temporarily registers it under the terminal ID as long as it is before a first expiration time T1 but after a reference time and provided no prior public key has already been temporarily registered for that TID. After T2, the temporarily registered public key becomes permanently registered, provided that no request to invalidate the TID was received prior to T2. The registered public key is used in all subsequent communications with the terminal. Consequently, only the terminal which generated the corresponding secret key can decrypt information received from the KDC or any other node that has obtained a copy of the registered public key of that terminal. Alternatively, it is possible to use the described protocol to register a secret key associated with a symmetric algorithm. In this case, a secret terminal key is generated at the terminal and sent without integrity, but encrypted under PKkdc, to the KDC, where it is recovered by decrypting under SKkdc.

The general exposure of the procedure is that an opponent can always initiate a successful sign-on from his location with his terminal, provided that the real terminal never signs on before T2, and does not report this to the KDC. In that case, the fake terminal can continue to operate indefinitely. But if the real terminal signs on before T1, the KDC will detect that a second initiation was tried, al-

though it cannot be determined which initiation is fake and which one is real. In such an event, the corresponding terminal will be taken out of the system and further investigations can be initiated. To reduce the exposure of a fake terminal being registered, the period for the registration of the public key (T1-T0) can be made as small as practically possible and the time parameters T0, T1 and T2 can be kept proprietary or secret.

From the foregoing, it will be appreciated that the invention provides a method of terminal initialisation which allows a remote terminal in a network to establish a common key or cryptovariable with a KDC where in advance the remote terminal has installed in it only the public key of the KDC. It will likewise be appreciated that the invention also provides a method of user initialisation which allows a user at a remote terminal in a network to establish a common user key or cryptovariable with a KDC where in advance the remote terminal has installed in it only the public key of the KDC and the identity of the user (UID). In this case, the key or cryptovariable generated at the terminal is associated with the user (UID) instead of the terminal (TID), and the user registers this key or cryptovariable under his UID instead of a TID. When a public key algorithm is involved, the user causes the terminal to generate a public key and secret key pair, where the public key is registered with the KDC under a UID and the secret key may be stored at the terminal or on a medium carried by the user from terminal to terminal; e.g. using a magnetic stripe card, a memory card, an intelligent secure card, a diskette or the like. When a symmetric algorithm is involved, the user causes the terminal to generate a secret key, which is registered with the KDC, under the UID. During transmission to the KDC, the so-generated secret key can be protected by encrypting it under the public key of the KDC (PKkdc). The secret key is also stored in the terminal or on a medium as described above. Ordinarily, each user registers his own cryptovariables with the KDC. Except for those obvious differences, the two procedures, for terminal and user initialisation, are the same.

Thus, while the invention has been described in terms of a preferred embodiment, those skilled in the art will therefore appreciate that the invention can be practised with modifications and variations within the scope of the appended claims.

Claims

1. A method of initialising cryptovariables in a multiterminal network which uses a public key algorithm and wherein the number of terminals is not essentially limited, the method comprising

establishing a key distribution centre within the network;

generating a public and secret key pair for the key distribution centre;

providing a cryptographic facility in each terminal that is intended to be secure;

storing the public key of the key distribution centre in each such cryptographic facility;

providing and storing at the key distribution centre, for each authorised entity - user or terminal - an individual identification; and

informing an initialiser - the user or an appointee in the case of a terminal - of two expiration times for the purposes of registering a cryptovariable with the key distribution centre in order that the initialiser can generate a cryptovariable at a or the terminal, using the cryptographic facility and transmit over the network to the key distribution centre a registration request message including the identification and the cryptovariable pair before the first of the expiration times;

the key distribution centre temporarily registering the cryptovariable against the identification, if received before the first expiration time and, if any further communication is received before the second expiration period in respect thereof, cancelling the registration, whereas, after the second expiration period, only cancelling the registration upon proof of authorisation.

2. A method as claimed in claim 1 wherein the request message transmitted to the key distribution centre includes time variant data and the step of temporarily registering the cryptovariable includes the step of acknowledging the temporary registration of the cryptovariable by sending a message to the requesting terminal containing the identification, the public key, the time variant data echoed, and a signature using the secret key of the key distribution centre.

3. A method as claimed in either preceding claim, further comprising the step of preparing and transmitting to the key distribution centre prior to the second expiration time a identification invalidation request message in the event that neither of a positive acknowledgement from the key distribution centre stating that the requested public key was temporarily registered at the key distribution centre and a positive acknowledgement from the key dis-

tribution centre stating that the requested identification has been invalidated by the key distribution centre is received.

4. A method as claimed in any preceding claim, further comprising the steps of: requesting the key distribution centre to provide a verification of the cryptovariable registration for the identification after the second expiration time has passed; and providing verification of the registration if the identification is valid and a cryptovariable was previously registered.

5. A method as claimed in claim 4 wherein the step of providing verification of the registration is provided by sending a message specifying the registered public key for that entity, the identification and a signature using the secret key of the key distribution centre.

6. A method as claimed in any preceding claim, wherein the public key of the key distribution centre is installed in each terminal at a central location before shipping the terminal to its point of use.

7. A method as claimed in any preceding claim, wherein the step of generating is performed by generating a public key and secret key pair for each authorised entity.

8. A method as claimed in any of claims 1 to 6, wherein the step of generating is performed by generating a secret authorised entity key using a symmetric algorithm.

9. A method as claimed in any preceding claim, further including proving the identification of the designated initialiser after the second expiration time has passed; and requesting the invalidation of the identification if the requested cryptovariable was not registered with the key distribution centre.

10. A method as claimed in any preceding claim, wherein, where an authorised entity is a user, the cryptovariable is recorded on a medium readable by terminals in the network.

FIG.1

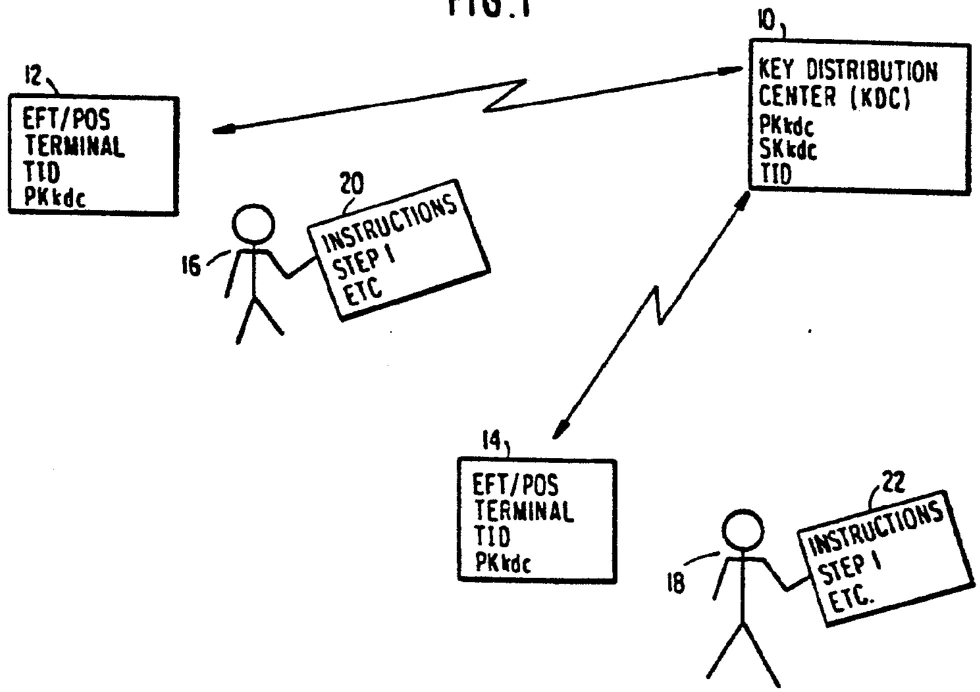
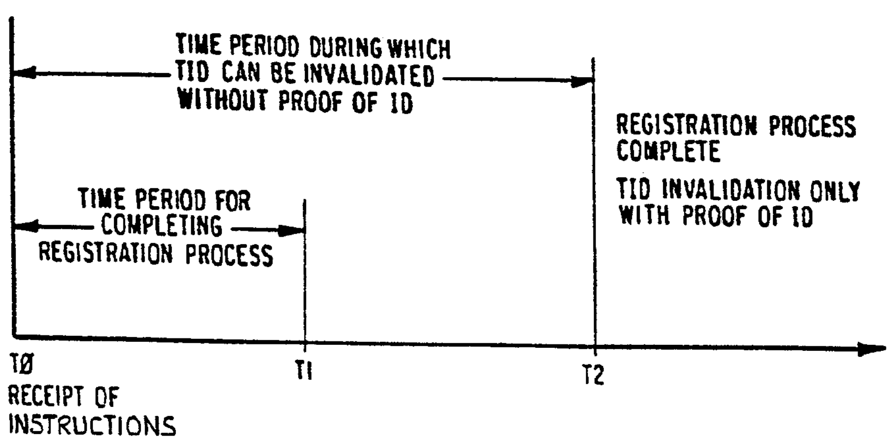

10
KEY DISTRIBUTION CENTER (KDC)
PKkdc
SKkdc
TID
12
EFT/POS TERMINAL
TID
PKkdc
20
INSTRUCTIONS
STEP 1
ETC
16
14
EFT/POS TERMINAL
TID
PKkdc
22
INSTRUCTIONS
STEP 1
ETC.
18

FIG.2


TIME PERIOD DURING WHICH TID CAN BE INVALIDATED WITHOUT PROOF OF ID
REGISTRATION PROCESS COMPLETE
TID INVALIDATION ONLY WITH PROOF OF ID
TIME PERIOD FOR COMPLETING REGISTRATION PROCESS
TØ
RECEIPT OF INSTRUCTIONS
T1
T2

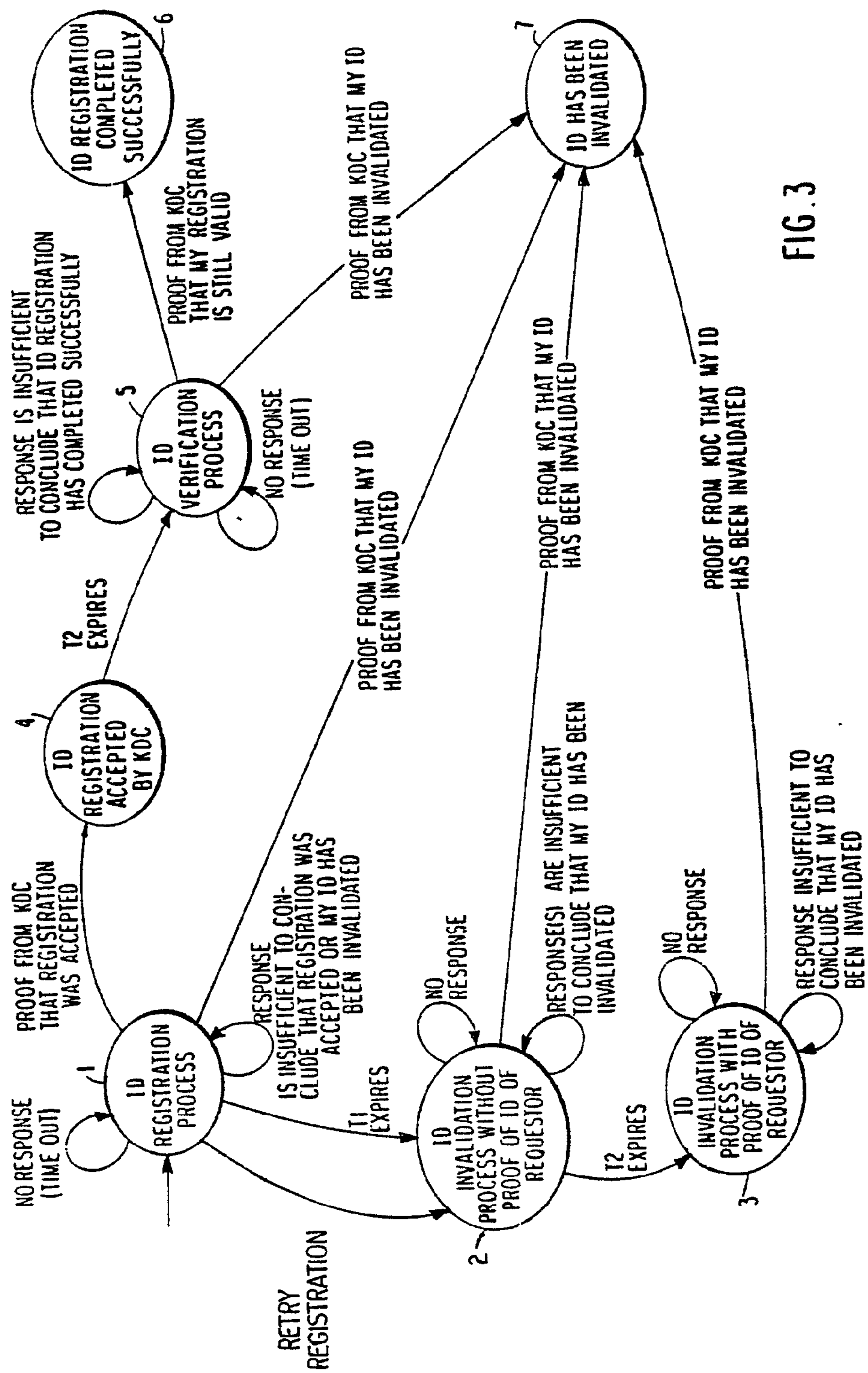

ID REGISTRATION PROCESS
1
NO RESPONSE (TIME OUT)
RETRY REGISTRATION
PROOF FROM KDC THAT REGISTRATION WAS ACCEPTED
RESPONSE IS INSUFFICIENT TO CONCLUDE THAT REGISTRATION WAS ACCEPTED OR MY ID HAS BEEN INVALIDATED
ID REGISTRATION ACCEPTED BY KDC
4
T2 EXPIRES
ID VERIFICATION PROCESS
5
RESPONSE IS INSUFFICIENT TO CONCLUDE THAT ID REGISTRATION HAS COMPLETED SUCCESSFULLY
NO RESPONSE (TIME OUT)
PROOF FROM KDC THAT MY REGISTRATION IS STILL VALID
ID REGISTRATION COMPLETED SUCCESSFULLY
6
PROOF FROM KDC THAT MY ID HAS BEEN INVALIDATED
ID HAS BEEN INVALIDATED
7
T1 EXPIRES
ID INVALIDATION PROCESS WITHOUT PROOF OF ID OF REQUESTOR
2
NO RESPONSE
RESPONSE(S) ARE INSUFFICIENT TO CONCLUDE THAT MY ID HAS BEEN INVALIDATED
PROOF FROM KDC THAT MY ID HAS BEEN INVALIDATED
PROOF FROM KDC THAT MY ID HAS BEEN INVALIDATED
T2 EXPIRES
ID INVALIDATION PROCESS WITH PROOF OF ID OF REQUESTOR
3
NO RESPONSE
RESPONSE INSUFFICIENT TO CONCLUDE THAT MY ID HAS BEEN INVALIDATED
PROOF FROM KDC THAT MY ID HAS BEEN INVALIDATED

FIG. 3

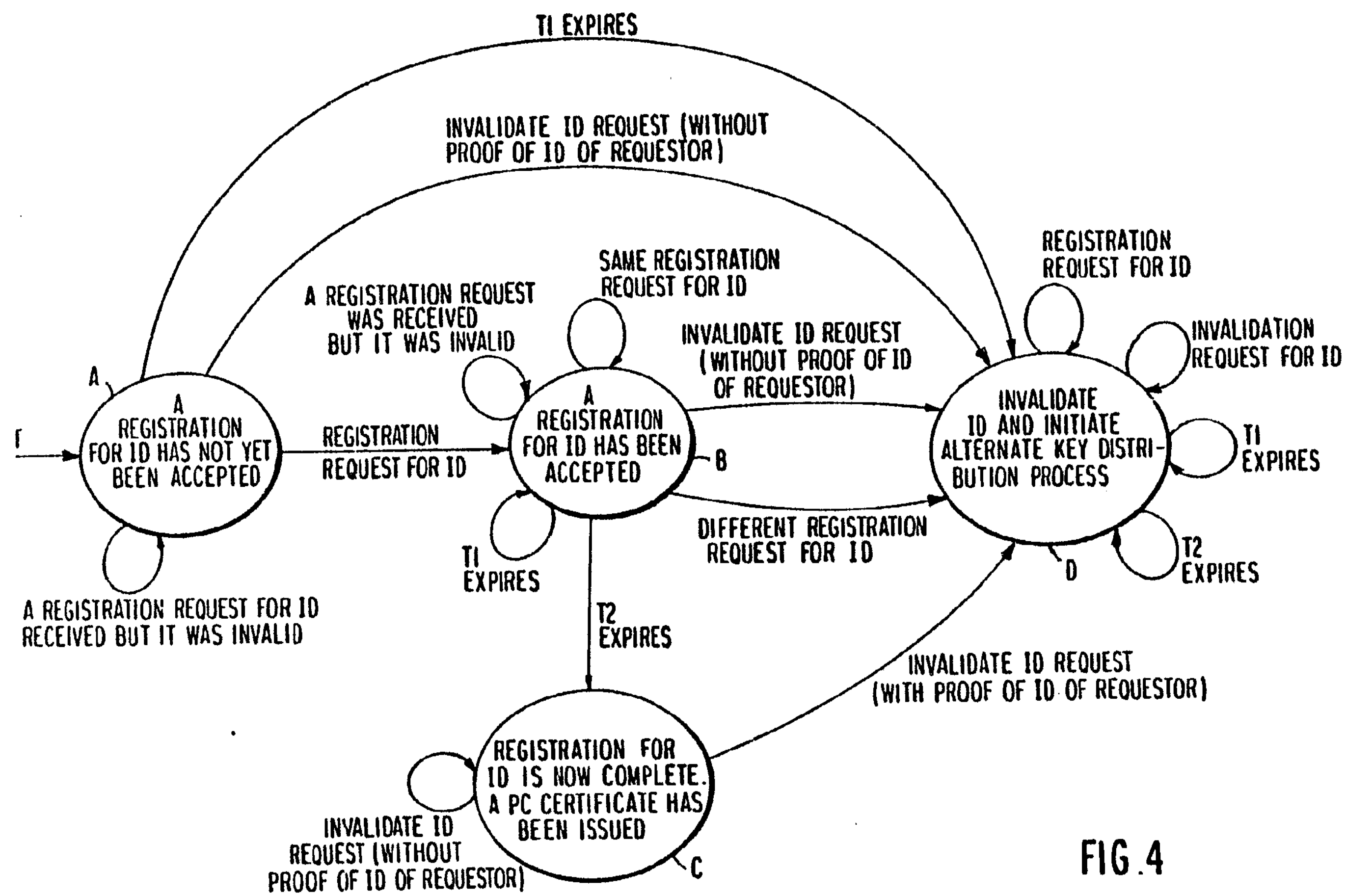

T1 EXPIRES
INVALIDATE ID REQUEST (WITHOUT PROOF OF ID OF REQUESTOR)
A
A REGISTRATION FOR ID HAS NOT YET BEEN ACCEPTED
A REGISTRATION REQUEST FOR ID RECEIVED BUT IT WAS INVALID
REGISTRATION REQUEST FOR ID
A REGISTRATION REQUEST WAS RECEIVED BUT IT WAS INVALID
SAME REGISTRATION REQUEST FOR ID
A REGISTRATION FOR ID HAS BEEN ACCEPTED
B
T1 EXPIRES
T2 EXPIRES
INVALIDATE ID REQUEST (WITHOUT PROOF OF ID OF REQUESTOR)
DIFFERENT REGISTRATION REQUEST FOR ID
REGISTRATION FOR ID IS NOW COMPLETE. A PC CERTIFICATE HAS BEEN ISSUED
C
INVALIDATE ID REQUEST (WITHOUT PROOF OF ID OF REQUESTOR)
INVALIDATE ID REQUEST (WITH PROOF OF ID OF REQUESTOR)
INVALIDATE ID AND INITIATE ALTERNATE KEY DISTRIBUTION PROCESS
D
REGISTRATION REQUEST FOR ID
INVALIDATION REQUEST FOR ID
T1 EXPIRES
T2 EXPIRES

FIG.4